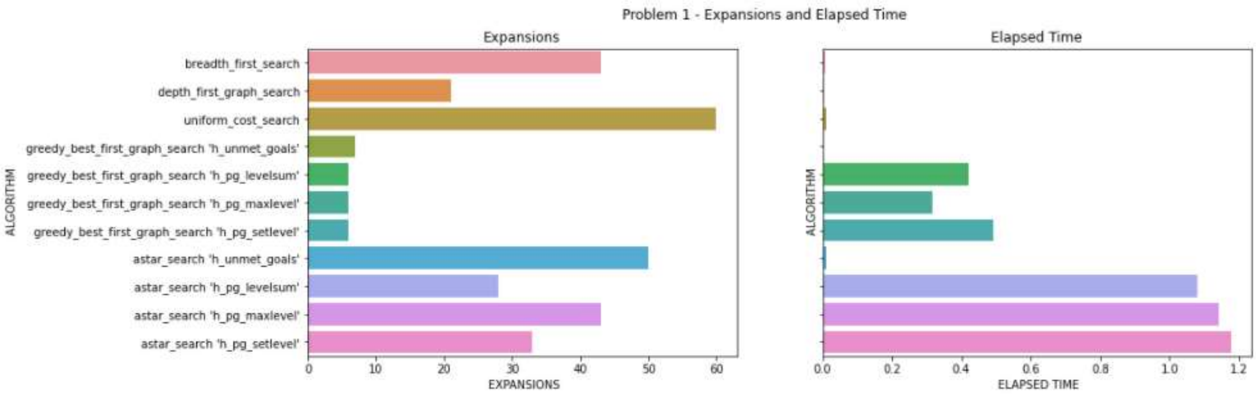
Project: Build a Forward-Planning Agent

Presenting the project results

Results for the algorithms applied in problem 1.

```
Problem1
```

	PROBLEM	ALGORITHM	ACTIONS	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	ELAPSED TIME
0	Air Cargo Problem 1	breadth_first_search	20	43	56	178	6	0.006309
1	Air Cargo Problem 1	depth_first_graph_search	20	21	22	84	20	0.003452
2	Air Cargo Problem 1	uniform_cost_search	20	60	62	240	6	0.010305
3	Air Cargo Problem 1	greedy_best_first_graph_search 'h_unmet_goals'	20	7	9	29	6	0.001748
4	Air Cargo Problem 1	greedy_best_first_graph_search 'h_pg_levelsum'	20	6	8	28	6	0.419876
5	Air Cargo Problem 1	greedy_best_first_graph_search 'h_pg_maxlevel'	20	6	8	24	6	0.315520
6	Air Cargo Problem 1	greedy_best_first_graph_search 'h_pg_setlevel'	20	6	8	28	6	0.492327
7	Air Cargo Problem 1	astar_search 'h_unmet_goals'	20	50	52	206	6	0.009890
8	Air Cargo Problem 1	astar_search 'h_pg_levelsum'	20	28	30	122	6	1.080640
9	Air Cargo Problem 1	astar_search 'h_pg_maxlevel'	20	43	45	180	6	1.141485
10	Air Cargo Problem 1	astar_search 'h_pg_setlevel'	20	33	35	138	6	1.177569

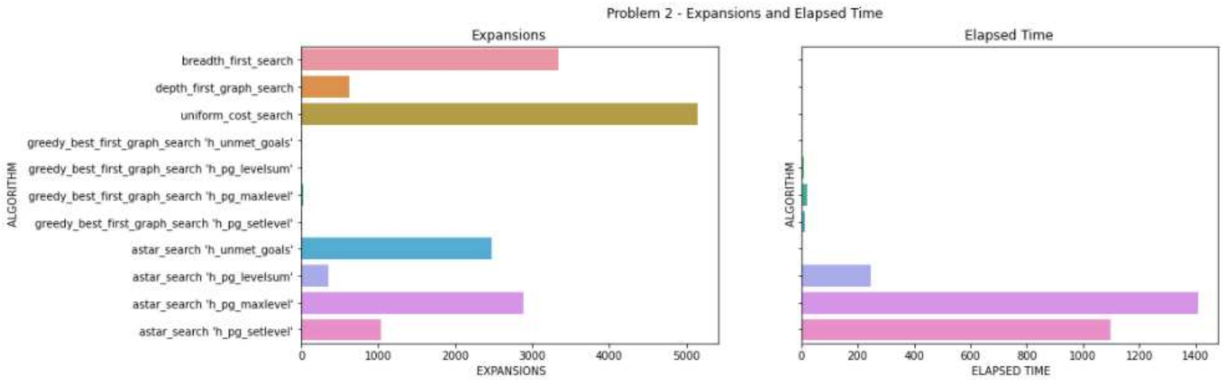


Analysis 1 : In the results of the first problem, it is possible to see that the uniform_cost_search algorithm had the largest number of expanded nodes and that greedy_best_first_graph_search had the lowest number of expanded nodes.

Results for the algorithms applied in problem 2.

```
Problem2
```

	PROBLEM	ALGORITHM	ACTIONS	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	ELAPSED TIME
0	Air Cargo Problem 2	breadth_first_search	72	3343	4609	30503	9	2.033861
1	Air Cargo Problem 2	depth_first_graph_search	72	624	625	5602	619	2.976749
2	Air Cargo Problem 2	uniform_cost_search	72	5154	5156	46618	9	3.369946
3	Air Cargo Problem 2	greedy_best_first_graph_search 'h_unmet_goals'	72	17	19	170	9	0.020013
4	Air Cargo Problem 2	greedy_best_first_graph_search 'h_pg_levelsum'	72	9	11	86	9	9.662020
5	Air Cargo Problem 2	greedy_best_first_graph_search 'h_pg_maxlevel'	72	27	29	249	9	19.605490
6	Air Cargo Problem 2	greedy_best_first_graph_search 'h_pg_setlevel'	72	9	11	84	9	12.366171
7	Air Cargo Problem 2	astar_search 'h_unmet_goals'	72	2467	2469	22522	9	2.333990
8	Air Cargo Problem 2	astar_search 'h_pg_levelsum'	72	357	359	3426	9	245.628787
9	Air Cargo Problem 2	astar_search 'h_pg_maxlevel'	72	2887	2889	26594	9	1408.171859
10	Air Cargo Problem 2	astar_search 'h_pg_setlevel'	72	1037	1039	9605	9	1097.932770



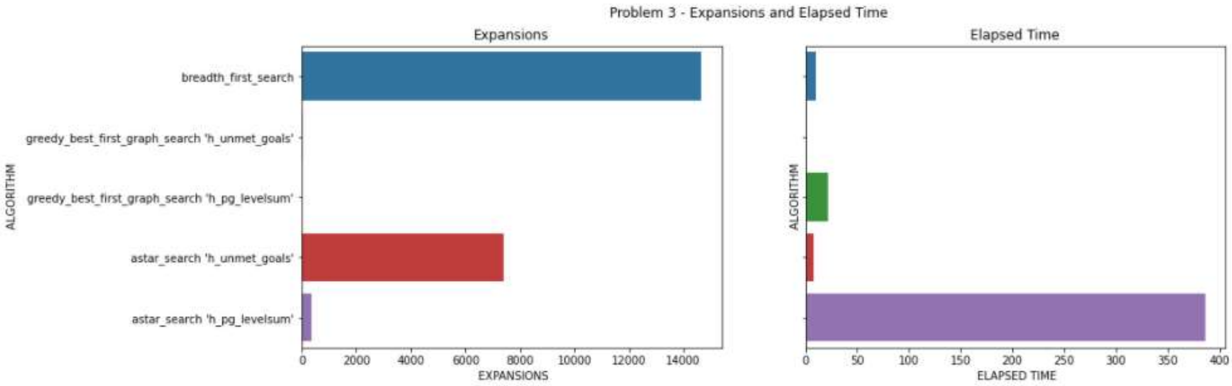
Analysis 2: In the results of the second problem, it is possible to see that the uniform_cost_search algorithm maintained the largest number of expanded nodes and that greedy_best_first_graph_search presented the lowest number of expanded nodes and a good relationship with time.

A comparison between the first two problems shows that the number of expanded nodes increases with increasing domain size.

Results for the algorithms applied in problem 3.

```
Problem3
```

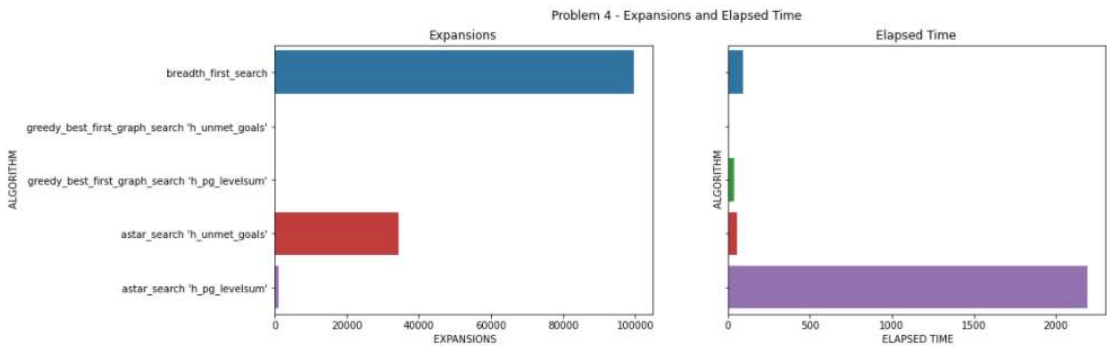
	PROBLEM	ALGORITHM	ACTIONS	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	ELAPSED TIME
0	Air Cargo Problem 3	breadth_first_search	88	14663	18098	129625	12	10.384480
1	Air Cargo Problem 3	greedy_best_first_graph_search 'h_unmet_goals'	88	25	27	230	15	0.037763
2	Air Cargo Problem 3	greedy_best_first_graph_search 'h_pg_levelsum'	88	14	16	126	14	21.301645
3	Air Cargo Problem 3	astar_search 'h_unmet_goals'	88	7388	7390	65711	12	8.329289
4	Air Cargo Problem 3	astar_search 'h_pg_levelsum'	88	369	371	3403	12	386.598955



Results for the algorithms applied in problem 4.

```
Problem4
```

	PROBLEM	ALGORITHM	ACTIONS	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	ELAPSED TIME
0	Air Cargo Problem 4	breadth_first_search	104	99736	114953	944130	14	94.127341
1	Air Cargo Problem 4	greedy_best_first_graph_search 'h_unmet_goals'	104	29	31	280	18	0.057786
2	Air Cargo Problem 4	greedy_best_first_graph_search 'h_pg_levelsum'	104	17	19	165	17	39.270902
3	Air Cargo Problem 4	astar_search 'h_unmet_goals'	104	34330	34332	328509	14	55.179819
4	Air Cargo Problem 4	astar_search 'h_pg_levelsum'	104	1208	1210	12210	15	2191.971979



Analysis 3: For problems 3 and 4, only three algorithms were chosen, two of which with two different heuristics. The greedy_best_first_graph_search algorithm had the least number of expanded nodes and maintained a good relationship with time, even with the increase in the domain size. The A* algorithm presented an interesting result for these domains in relation to the expanded nodes, but it presented a high cost in relation to time. It was found that the chosen heuristic had a great influence on the result in relation to the A* algorithm.

Questions

Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

From the verified performances, I believe that the most suitable algorithm for planning in a very restricted domain and that needs to operate in real time would be "greedy_best_first_graph_search 'h_unmet_goals'".

Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

Considering the results in problems 3 and 4, I think that the algorithms that would be more appropriate for planning in very large domains would be: "greedy_best_first_graph_search 'h_unmet_goals'", "greedy_best_first_graph_search 'h_pg_levelsum'", "astar_search 'h_unmet_goals' ". The choice would depend on the other characteristics of the problem.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

For planning problems where it is important to find only optimal plans, I think that the A * algorithm is very good, as long as there is an allowable heuristic. The Breadth-first search is also complete and great for unit cost steps, but it has the complexity of exponential time.